

# Sistemas Operativos

## Tarea 3:

Profesor: Viktor Tapia  
Ayudante Cátedra: Joaquín Castillo  
Ayudante Tareas: Giorgio Pellizzari

13 de Noviembre, 2020

## 1 Reglas Generales

Para la siguiente tarea se debe realizar un código programado en lenguaje Java. Se exigirá que los archivos se presenten de la forma más limpia y legible posible. Deberá incluir un archivo README con las instrucciones de uso de sus programas junto a cualquier indicación que sea necesaria, y un archivo MAKE para poder ejecutar el programa.

## 2 Tarea

Se le solicita resolver los siguientes problemas utilizando threads.

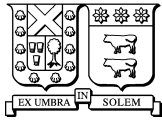
### 2.1 Problema 1:

Deberá generar un programa que resuelva ecuaciones de funciones dependientes entre si. El programa recibirá de input un archivo de nombre *funciones.txt* en el cual se indicarán el numero de funciones, seguido de las funciones a utilizar. Estas funciones poseerán operaciones de suma, resta, multiplicación y división siguiendo sus respectivos ordenes de precedencia. Un ejemplo del archivo *funciones.txt* es el siguiente:

```
3
f(x)=g(x)*h(x)
g(x)=(h(x)+1)/2
h(x)=2*x+1
```

Un ejemplo del funcionamiento del programa es el siguiente:

```
Funciones ingresadas!
Ingrese operacion:
>f(1)
El resultado es: 6
Ingrese operacion:
>
```



El programa deberá generar un thread para resolver la ecuación. En caso de tener funciones anidadas, deberá generar nuevos threads que resuelvan las funciones anidadas (un thread para cada función). Esto se deberá hacer hasta llegar a una función que no dependa de otras, retornando el resultado al thread que solicitó su cálculo, ya sea este el original, o uno que haya necesitado generar threads para resolver funciones.

## 2.2 Problema 2:

Deberá comparar el rendimiento (en tiempo) de dos algoritmos de ordenamiento sobre un set de arreglos de números aleatorios. Los algoritmos deben ser iguales, con la diferencia de que uno este implementado con threads y el otro no. Deberán hacer su propio set de datos para comparar siguiendo el formato que mas les acomode, esto con el fin de que puedan notar diferencias entre los algoritmos con mayor facilidad.

Justifique en el readme las decisiones tomadas en la implementación y determine si el uso de threads beneficia o no a este tipo de algoritmos.

Pueden apoyarse de cualquier algoritmo ya conocido por usted o usar implementaciones de otros cursos para apoyarse.

## 3 Consideraciones para la entrega

- Se deberá trabajar de a pares. Se deberá entregar en Aula a mas tardar el día 4 de Diciembre del 2020 a las 23:55 horas. Se descontarán 5 puntos por cada hora o fracción de atraso. Las copias serán evaluadas con nota 0 en el promedio de las tareas.
- La tarea debe ser hecha en el lenguaje Java.
- Los archivos deberán ser comprimidos y enviados juntos en un archivo .tar.gz en el formato TAREA3.ROL1.ROL2.
- Las preguntas deben ser hechas por *Aula*
- Se **descontarán** 30 puntos por:
  - Mala redacción en el README (o no hay).
  - Mala implementación del makefile (o no hay).
  - No respetar el formato de entrega.
  - Tarea en borrador.

## 4 Presentaciones

Para cada tarea, se seleccionarán grupos al azar para presentar su tarea frente a ayudantes y eventualmente profesor, recibiendo una ponderación del 75% y 25% entre tarea y presentación respectivamente. Si su grupo presentó en una tarea, no volverá a salir nuevamente.

Las presentaciones deberán tener como mínimo los siguiente:

- Explicar de qué trata la tarea.
- Mostrar la tarea funcionando con casos de prueba de las condiciones para ver su funcionalidad.
- Explicar brevemente su implementación.